Journal of Power Sources 269 (2014) 682–693

Contents lists available at ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

State-of-charge estimation for battery management system using optimized support vector machine for regression

J.N. Hu [a], J.J. Hu [b], H.B. Lin [a], X.P. Li [a, *], C.L. Jiang [a], X.H. Qiu [a], W.S. Li [a, *]

[a] School of Chemistry and Environment, Key Laboratory of Electrochemical Technology on Energy Storage and Power Generation of Guangdong Higher Education Institutes, and Engineering Research Center of Materials and Technology for Electrochemical Energy Storage (Ministry of Education), South China Normal University, Guangzhou 510006, China

[b] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

HIGHLIGHTS

- Utilizes double step search for selecting the parameters *c* and *g* for SVR with RBF kernel.
- Optimized SVR exhibits generalization ability to samples under different running conditions.
- Optimized SVR fits the nonlinear curve of battery capacity with less MSE and error than ANN.

ARTICLE INFO

Article history:
Received 28 December 2013
Received in revised form
30 June 2014
Accepted 3 July 2014
Available online 11 July 2014

Keywords:
State of charge
Support vector machine for regression
Battery management system
Electric vehicle
Double step search
Driving conditions

ABSTRACT

State-of-charge (SOC) estimation is one of the most challengeable tasks for battery management system (BMS) in electric vehicles. Since the external factors (voltage, current, temperature, arrangement of the batteries, etc.) are complicated, the formula of SOC is difficult to deduce and the existent SOC estimation methods are not generally suitable for the same vehicle running in different road conditions. In this paper, we propose a new SOC estimation based on an optimized support vector machine for regression (SVR) with double search optimization process. Our developed method is tested by simulation experiments in the ADVISOR, with a comparison of the estimations based on artificial neural network (ANN). It is demonstrated that our method is simpler and more accurate than that based on ANN to deal with the SOC estimation task.

1. Introduction

Energy shortage and environmental pollution problems motivate people to develop electric vehicles with efficient energy management [1–3]. The key component of an electric vehicle is the battery, which determines not only the running distance but also the cost of an electric vehicle. Researchers have been interested in exploring new electrochemical systems to improve battery performance. Whatever electrochemical system is adopted, battery management system (BMS) is indispensable to control a battery pack, especially for lithium ion battery, which is more sensitive to misusages [3–7]. Misusages, such as overcharge and overdischarge, seriously shorten the life time of the battery [8,9]. BMS can effectively prevent the battery misusages with the accurate estimation of state-of-charge (SOC). Therefore, SOC estimation is one of the most important tasks for BMS [10–14]. Various methods have been proposed to estimate SOC.

Open circuit voltage (OCV) measurement is the most straightforward method for SOC estimation. The estimated SOC values from OCV can be approximately regarded as the true SOC value, only when a linear relational mapping from OCV to SOC ideally exists. Unfortunately, this situation seldom happens for most batteries. Even if there is a linear relationship, this relationship will be impacted by external conditions such as

* Corresponding authors. School of Chemistry and Environment, South China Normal University, Guangzhou 510006, China. Tel./fax: +86 020 39310256.
E-mail address: liwsh@scnu.edu.cn (W.S. Li).

http://dx.doi.org/10.1016/j.jpowsour.2014.07.016
0378-7753/

temperature [15]. Besides, the accurate voltage value of the battery pack for electric vehicles can be collected at parking state but cannot at working states.

Current integral method [16] was then developed for the SOC estimation and has been widely used in various battery systems. Compared to the OCV measurement, current integral method can estimate SOC in the driving process. However, the key point of current integral is to collect the current values constantly and accurately. Once the current values are collected inaccurately, the accumulation of the current error leads to the poor SOC estimation. Although the current integral method is simple and easy to be implemented, the external conditions, such as jolt in the driving process, external temperature and the humidity, usually affect the current collection. To improve the accuracy of SOC estimation from current integral, a learning method [17] has been proposed for estimating the SOC curve via learning the non-linear relationship from other indicators such as temperature and charge–discharge rate. Obviously, the learning algorithm is crucial in SOC estimation and the accuracy of SOC estimation can be further improved by using the better algorithm.

The Kalman filtering method is one of the classical algorithms to reduce the noise in the signals. Dai et al. proposed an extended Kalman filtering method for SOC estimation, which reduces the computation and improves the accuracy of SOC estimation [18]. The key point of Kalman filtering method is the optimal estimation for power system in minimum variance calculation [11–14]. With this advantage, the SOC estimation based on Kalman filtering method is suitable for various kinds of battery [19]. Kalman filtering method can estimate the SOC value from the incomplete and noisy data, but this algorithm is complicated.

The accuracy of the SOC estimation could be further improved by artificial neural network (ANN) [20,21]. With a collection of the different indicators for a training process of ANN, a satisfactory SOC estimation with a high correlation between SOC and indicators was obtained [20]. Because of its nonlinearity, ANN can build up a neural architecture by neurons and hidden layers to approximate the dynamic properties of the battery, which is usually composed of the input layer, the hidden layers and the output layer. However, the training process of the neural network is too complex, involving the type of the transfer function, the number of the hidden neurons and the hidden layers, and is not suitable for real embedded systems. Additionally, an appropriate activated function reflecting the relation between the output of neurons and the activated states should be selected empirically based on a huge number of data, which is time-consuming and leads to overfitting.

In this paper, we proposed a new SOC estimation based on an optimized support vector machine for regression (SVR) with double search optimization process. SVR provides a hyperplane of high dimensionality to approximate the training data and demonstrates its good performance in estimating the SOC for the lithium-ion and lithium-ion-polymer battery pack [23,24]. When it was evaluated using the cycle conditions in ADVISOR, our method behaved more efficiently and accurately than those based on ANN. Since the cycle conditions that ADVISOR provides are universal, our method is applicable to all kinds of batteries including lead-acid, metal hydride–nickel and lithium ion batteries.

2. Principle and methodology

As defined by United States Advanced Battery Consortium (USABC), SOC of battery is the ratio of the remaining charge (C_r) to the normal charge (C_n):

$$\mathrm{SOC} = \frac{C_r}{C_n} \times 100\%$$

The charge state of the battery where power can no more flow into the battery is generally defined as 100%, while the charge state of the battery where the power is used up is defined as 0%.

In this session, the background for SVR is given first and then our training method for SVR is introduced in detail.

2.1. Support vector machine for regression (SVR)

Support Vector Machine is first designed to solve the nonlinear two-class classification problem [22]. The key point of Support Vector Machine is to project the original sample ($\boldsymbol{x}$) from the low dimensional space ($\mathbb{R}^m$) to a high dimensional space ($\mathbb{R}^n$) such that a linear hyperplane can be found to separate samples from two classes, as shown in Fig. 1. Hence every nonlinear separable problem can be solved after mapping to a high or even infinitely high dimensional space by the kernel function [22,25]. We can train the Support Vector Machine for Regression (SVR) to predict a physical parameter [25]:

$$f(\boldsymbol{x}) = \boldsymbol{w}\phi(\boldsymbol{x}) + b, \boldsymbol{w} \in \mathbb{R}^n, \boldsymbol{x} \in \mathbb{R}^m, b \in \mathbb{R} \tag{1}$$

where $\boldsymbol{x}$, $\boldsymbol{w}$, b are the input data in m-dimensional space, the weight of the hyperplane in n-dimensional space and the intercept of the hyperplane respectively. $\phi: x \rightarrow \mathscr{H}$ is a non-linear mapping function from input space $\mathbb{R}^m$ to high dimensional space $\mathbb{R}^n$ (n is much larger than m). $\mathscr{H}$ is a Reproducing Kernel Hilbert Space (RKHS), where dot product of the two elements can be approximated by a kernel function:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle_{\mathscr{H}} \tag{2}$$

Vapnik et al. [22] proposed that the kernel function under Mercer condition could be used to approximate the dot product of two elements in RKHS and thus greatly reduce the computational complexity. In this way, we can further express the weight in Eq. (1) as the combination of α_j and $\phi(\boldsymbol{x}_j)$, namely $\boldsymbol{w} = \sum_{j=1}^{l} \alpha_j y_j \phi(\boldsymbol{x}_j)$. Then by substituting the dot product of $\phi(\boldsymbol{x}_j)$ and $\phi(\boldsymbol{x})$ by kernel function in Eq. (2), we can convert Eqs. (1)–(3):

$$\begin{aligned} f(x) &= \sum_{j=1}^{l} \alpha_j y_j \langle \phi(\boldsymbol{x}_j), \phi(\boldsymbol{x}) \rangle_{\mathscr{H}} + b \\ &= \sum_{j=1}^{l} \alpha_j y_j K(\boldsymbol{x}_j, \boldsymbol{x}) + b \end{aligned} \tag{3}$$

where α_j is a positive value that measures the weight for the jth support vector $\boldsymbol{x}_j$ and y_j is the corresponding output value of the jth support vector $\boldsymbol{x}_j$.

There are several kernel functions, including linear kernel, polynomial kernel, radial basis function (RBF) kernel and Sigmoid kernel. Due to its good generalization ability [27], RBF kernel

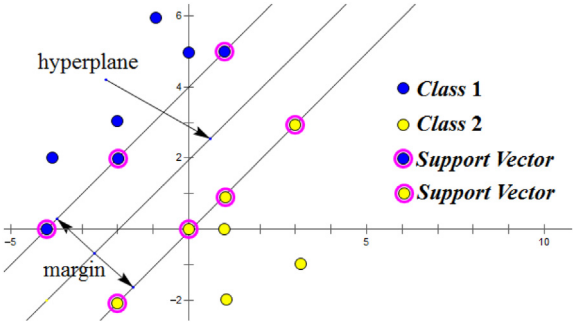

Fig. 1. Hyper plan to separate samples from two classes.

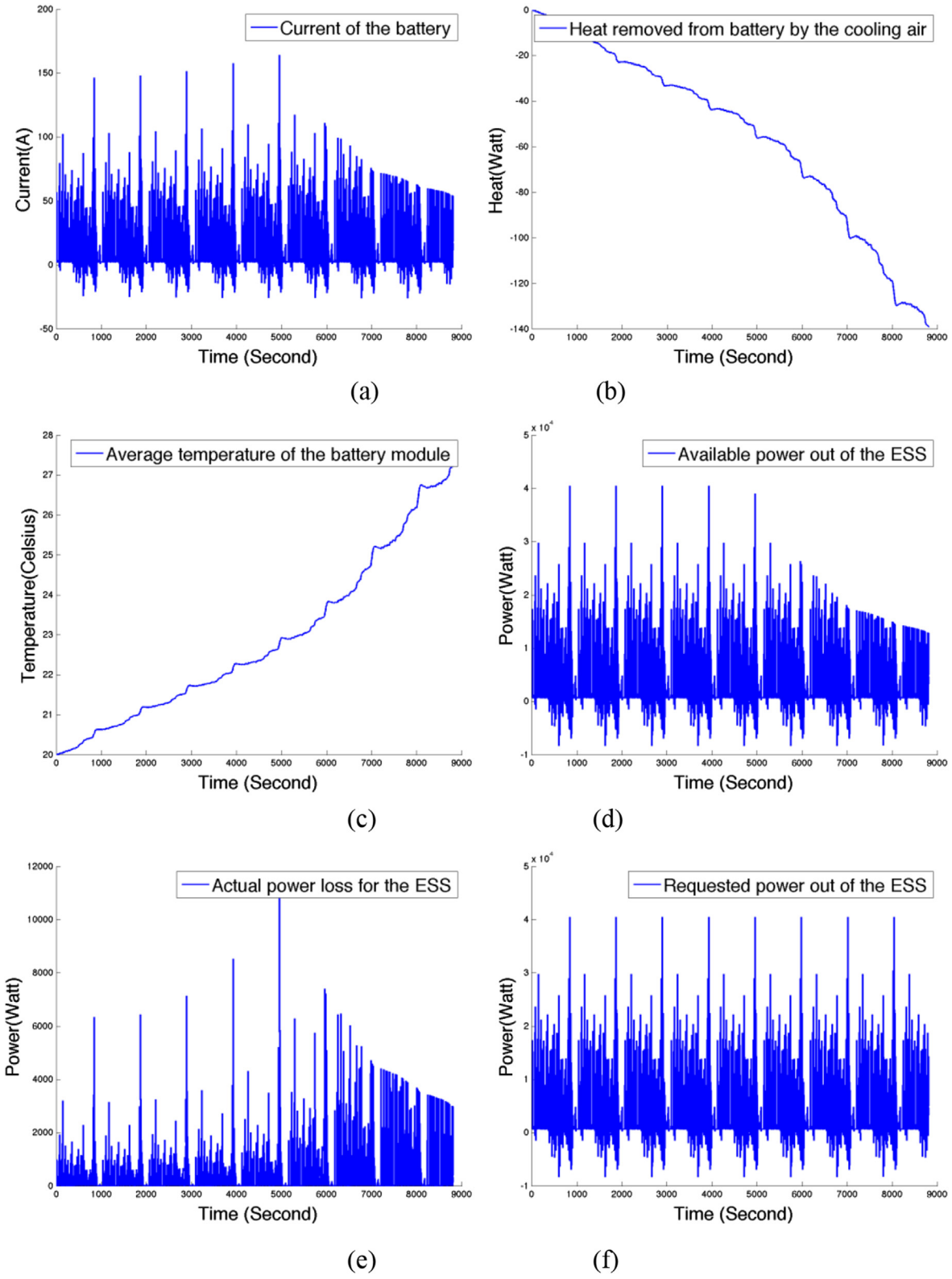

Fig. 2. Some collected features in the NYCCOM training dataset. (a) Current of the battery; (b) heat removed from the battery by the cooling air; (c) average temperature of battery module; (d) available power out of the ESS; (e) actual power loss for the ESS; (f) requested power out of the ESS.

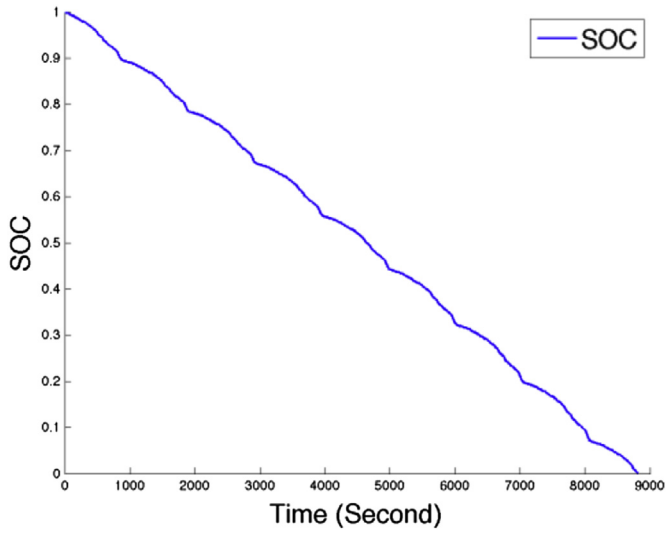

Fig. 3. The actual SOC value for the NYCCOM training dataset.

function, which is known as Gaussian kernel, is used in this paper, as Eq. (4):

$$K(\boldsymbol{x}_j, \boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_j\|^2}{2\sigma^2}\right) \tag{4}$$

where σ is the standard deviation of the Gaussian kernel. A.J. Smola et al. extended the concept of insensitive loss function based on the support vector machine to curve regression [25]. The insensitive loss function is defined as Eq. (5):

$$\ell(f(\boldsymbol{x}), y) = \begin{cases} 0, & |y - f(\boldsymbol{x})| \le \varepsilon \\ |y - f(\boldsymbol{x})| - \varepsilon, & |y - f(\boldsymbol{x})| > \varepsilon \end{cases} \tag{5}$$

where $f(x)$ is the predicted value of $\boldsymbol{x} \in \mathbb{R}^m$ given by f and ε is the relaxed threshold of the error between the predicted value and the true value $y \in \mathbb{R}$. The selection of hyper-parameters for SVR was further proposed in Ref. [26].

The training process of the SVR is to find the hyperplane so that the accumulated distance of all the training samples from the hyperplane is minimized. The linear regression function in the high dimensional space is defined in the same form as Eq. (1).

By introducing the slack variables ξ_i and ξ_i^*, the optimization problem can be represented as:

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$
$$\text{s.t.} \begin{cases} y_i - \boldsymbol{w}\phi(\boldsymbol{x}_i) - b \le \varepsilon + \xi_i \\ -y_i + \boldsymbol{w}\phi(\boldsymbol{x}_i) + b \le \varepsilon + \xi_i^* \quad , i = 1, 2, \ldots, l \\ \xi_i \ge 0, \xi_i^* \ge 0 \end{cases} \tag{6}$$

By means of the Lagrange duality, we can deduce the dual problem of the Eq. (6) and find an optimal solution of $\boldsymbol{w}$ and $\boldsymbol{b}$ for Eq. (1).

2.2. SOC estimation

In this paper, we used RBF kernel to find the optimal parameters for the SVR. The performance of SVR curve fitting can be greatly affected by some parameters, like optimal cost parameter c and g in the RBF kernel of SVR, where $g = -1/2\sigma^2$. In practice, electric vehicles are running in various driving conditions, such as sudden turns, straight lines, rugged road, etc. Hence under different cycle conditions, the electric vehicles consume different amount of the power, remove different amount of the heat and generate different current in the energy storage system (ESS). Optimal values of c and g are expected to be found when the mean square error (MSE) is the minimal of the all tests. Classical grid search follows the three steps: a) Change the parameters c and g each time in a wide range; b) Train a SVR using each pair of c and g each time; c) Compare the performance of SVR using different pairs of c and g and select the best SVR. However, training the SVR with a large number of data collected in different road conditions for different pairs of c and g by simple grid search is time-consuming. Hence we collect the representative training samples from several common driving conditions and apply a double step search to selecting the optimal factors, c and g.

The main purpose of the double step search is to accelerate the training process with searching purposefully and to approach the optimal parameters at the same time. Firstly, a rough search by a large searching step is applied to find the optimal pairs (C, G). It is expected that more optimal parameters can be found among (C, G) in a smaller region. Then a further search around (C, G) by a smaller searching step is applied to ultimately find the optimal pairs (*Best_C*, *Best_G*). Since the sample features are different in magnitude, all the features are normalized within the interval of [0, 1] before training the SVR estimator. The training algorithm is represented in the following pseudocode.

Pseudocode

1) Collect the original training samples by the simulation experiments on different driving conditions using ADVISOR;
2) Normalize all the training samples and form the representative training set T;
3) Loop by a large search step until all the possible pairs (c, g) in a large search space are tested
- Train SVRs using T and every possible (c, g);
- Compute MSE for training samples;
- Find the best pair (C, G) where MSE is minimal.

4) Loop by a smaller search step in the smaller search space around (C, G)
- Train SVRs using T and every possible (c, g);
- Compute MSE for training samples;
- Find the best pair (*Best_C*, *Best_G*) where MSE is minimal.

5) Get the optimal SVR trained by (*Best_C*, *Best_G*);
6) Estimate the SOC using the optimal SVR.

End

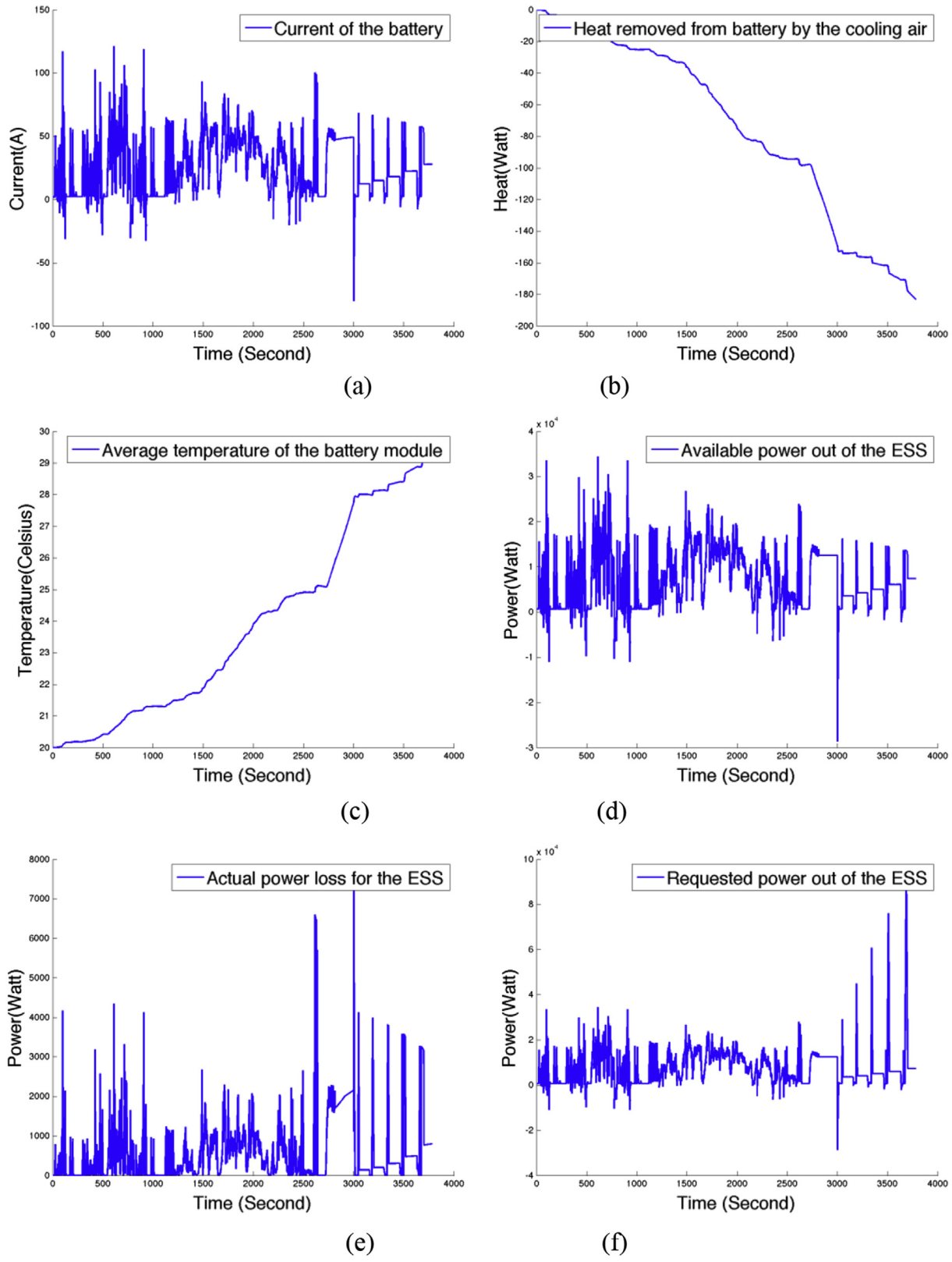

Fig. 4. Some collected features in the Combination 1 testing dataset. (a) Current of the battery; (b) heat removed from the battery by the cooling air; (c) average temperature of battery module; (d) available power out of the ESS; (e) actual power loss for the ESS; (f) requested power out of the ESS.

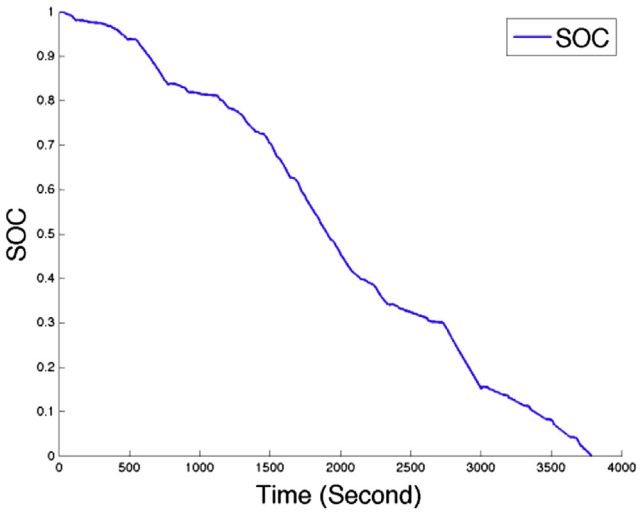

Fig. 5. The actual SOC value for the Combination 1 testing dataset.

3. Results and discussion

3.1. Features in datasets

To evaluate the accuracy of this method, we use the 8 common cycle conditions (ARTERIAL, NYCCOM, UDDSHDV, COMMUTER, WVUINTER, 5PEAK, CSHVR, CBD14) in ADVISOR. In this experiment, to estimate the current SOC value, we used the following seven data as the features: current out of the battery, module temperature of the energy storage system (ESS), available power out of the ESS, requested power out of ESS, actual power loss for the ESS, heat removed from battery by cooling air, and the average temperature of the cooling air.

In the first experiment, the training samples are collected when the same vehicle is repeatedly running in the driving conditions in ARTERIAL and NYCCOM separately until the charge of the battery is used up. Some of the sample features are shown in Fig. 2 and sample labels (SOC value) are showed in Fig. 3.

In the second experiment, the test data are collected when the same vehicle is simulated in 2 combined cycles (Combination 1 and Combination 2, see Fig. 4 and Fig. 5) until its charge is used up. Combination 1 consists of UDDSHDV, COMMUTER, WVUINTER and 5PEAK, while Combination 2 consists of UDDSHDV, CSHVR, COMMUTER and CBD14. We also test our optimized SVR to predict the SOC value when the car is running in 8 single cycles (UDDSHDV, CSHVR, COMMUTER, CBD14, WVUINTER, 5PEAK, ARTERIAL, NYCCOMP).

3.22. Performance

Seven classical neural networks, with different number of hidden neural, different number of hidden layers and different kinds of transfer functions, are also used for a comparison.

Back propagation neural network (BPNN) is a popular neural network which applies the classical back propagation method to update the weights between each layer. Diversified architectures (see Fig. 6a–e) are used in this experiment.

Generalized regression neural network (GRNN), which is often used for regression, is a kind of radial basis network with bias in the first layer. With radial basis neurons in the first layer, GRNN calculates the weighted sum of the input features with biases in the first layer. In the second layer, GRNN uses the linear transfer function, and calculates the weighted sum of the input features with the normalized dot product weight function (see Fig. 6f).

Elman network consists of 3 layers using the weight function, NETSUM function and the specified transfer functions, respectively. All the 3 layers have the biases in the weighted sum formula. The inputs are the weights for the first layer. Each subsequent layer has a weight coming from the previous layer. The last layer is the network output (see Fig. 6g).

Radial basis function neural network (RBFNN) is a network with biases in both of the two layers. RBFNN uses the radial basis function in the hidden layer and linear function in the output layer. In the hidden layer, RBFNN calculates the weighted sum of the input features with biases. In the second layer, RBFNN calculates the weighted input by dot product and its net inputs by NETSUM function (see Fig. 6h).

Figs. 7 and 8 show the result of the rough search in the double search process, while Figs. 9 and 10 show the second step. In Figs. 7 and 9, the training processing chooses the MSE in the z axis as the evaluation criterion, which means to find the lowest point in the surface. Figs. 8 and 10 are the MSE contours for Figs. 7 and 9, respectively. From the results above, we can see that double step search in each step finds the optimal solution to the objective function under MSE equation. This search narrows the searching space in each step and thus takes less time than searching globally with small steps in the c-g space.

The estimated results of the Combination 1 and Combination 2 are shown in Figs. 11 and 12, respectively. The error analysis by the optimized SVR method is show in Figs. 13 and 14. In Figs. 11 and 12, the estimated SOC by the optimized SVR best fits with the actual SOC curve with tiny error. As shown in Fig. 11a and b, the estimated performance is improved as the number of hidden neurons increases. It can be seen from Fig. 11a and c that the performance of BPNN varies with different transfer functions, and similar result can be seen in Fig. 11d and e. Log-sigmoid function based BPNN performs much better than tangent-sigmoid function based BPNN, but Log-sigmoid function based DBPNN is slightly inferior to tangent-sigmoid function based DBPNN. As shown in Fig. 11a and e, the estimated SOC curve fits slightly better as the number of the hidden layers increases. The ELMAN estimated curve fluctuates severely in the running process while GRNN have poor estimated outputs from time to time. By means of the same RBF method, the performance of SVR with RBF kernel is obviously better than RBFNN. The experiment result shows that the performance of the artificial neural networks is sensitive to their architectures in terms of the transfer function, number of the hidden neurons, number of the hidden layers, etc. Contrary to the complex architecture of ANN, SVR has fewer parameters to be tuned and has better estimation performance in this scenario.

The MSE evaluations for the 10 cycles conditions are recorded in Table 1 and the error sums for the 10 cycles conditions are recorded in Table 2. In Tables 1 and 2, double search based SVR performs the best in 9 cycle conditions except for the CBD14. Trained by the ARTERIAL and NYCCOM dataset, optimized SVR presents better generalization ability to unseen cycle conditions than the other methods. The MSE and error sum of our proposed SVR method are much lower than that of the other estimations.

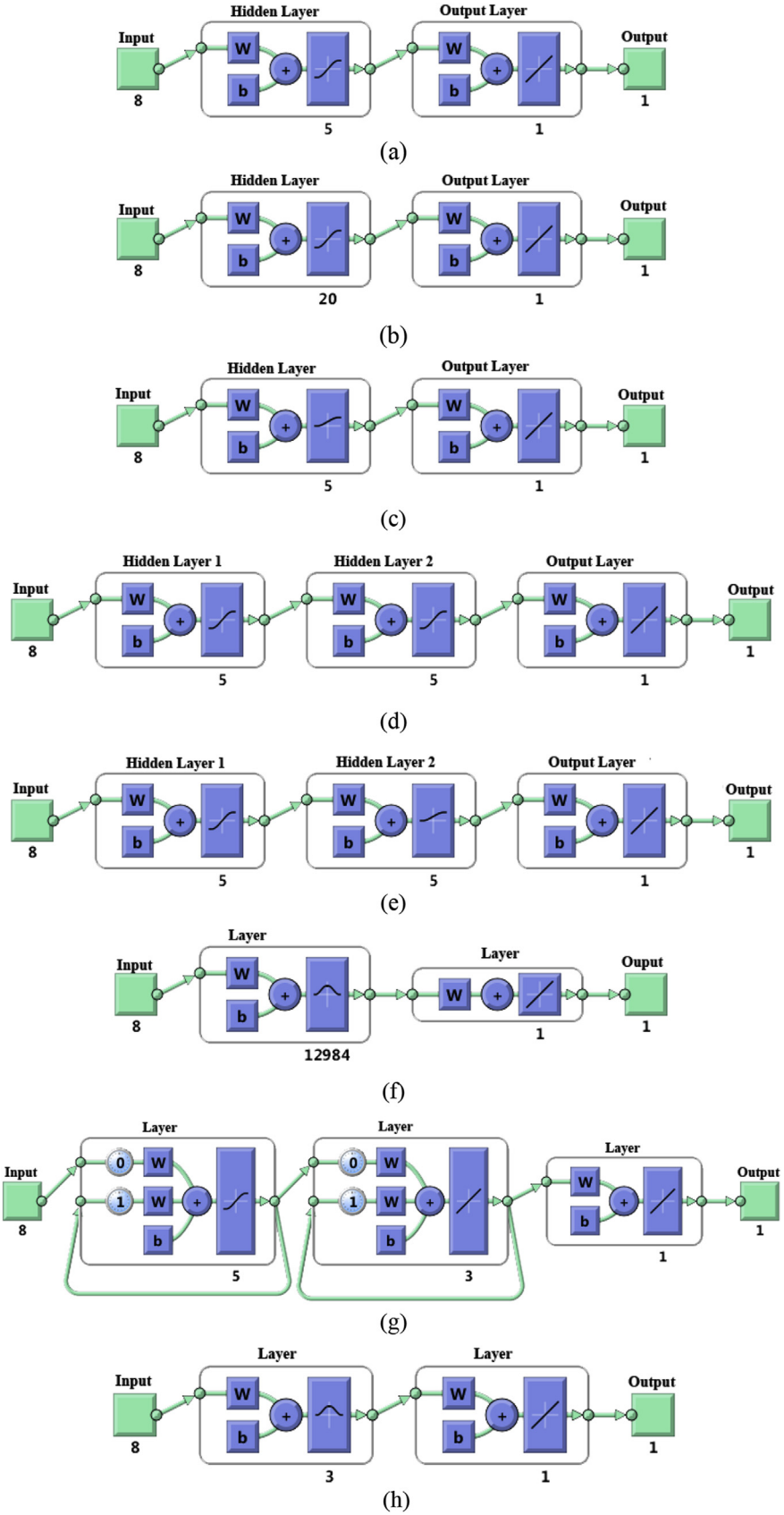

Fig. 6. (a) BP1: 5 hidden neurons using the hyperbolic tangent sigmoid transfer function (tansig) in only one hidden layer, one output using the pure linear output function in the output layer; (b) BP2: 20 hidden neurons using the hyperbolic tangent sigmoid transfer function (tansig) in only one hidden layer, one output using the pure linear output function in the output layer; (c) BP3: 5 hidden neurons using the log-sigmoid transfer function (logsig) in only one hidden layer, one output using the pure linear output function in the output layer; (d) DBP1: 5 hidden neurons using the hyperbolic tangent sigmoid transfer function (tansig) in each of the two hidden layers, one output using the pure linear output function in the output layer; (e) DBP2: 5 hidden neurons using the tan and log-sigmoid transfer function respectively in the two hidden layers, one output using the pure linear output function in the output layer; (f) GRNN: 12,984 hidden neurons using radial basis transfer function with the weights and biases, one output using the linear output function without biases in the output layer; (g) ELMAN: 5 hidden neurons using the hyperbolic tangent sigmoid transfer function (tansig) for the first hidden layer, 3 hidden neurons for the second hidden layer, one output using the linear output function in the output layer; (h) RBFNN: 3 hidden neurons using the radial basis function, one output using the linear output function in the output layer.

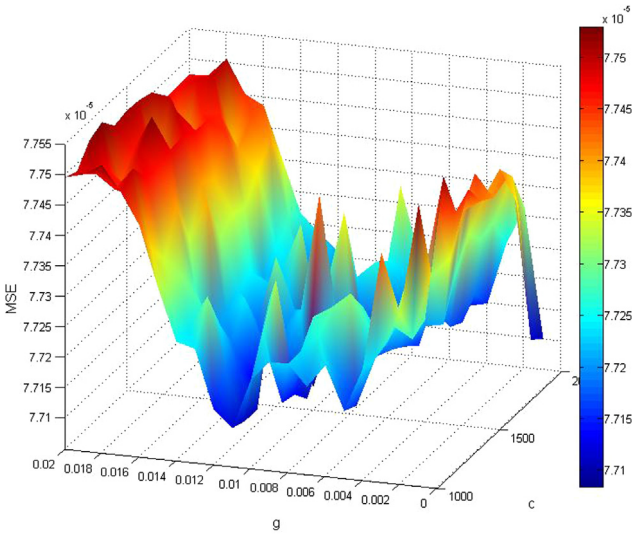

Fig. 7. Rough search for $c \in (1000,1500)$ and $g \in (0,0.02)$.

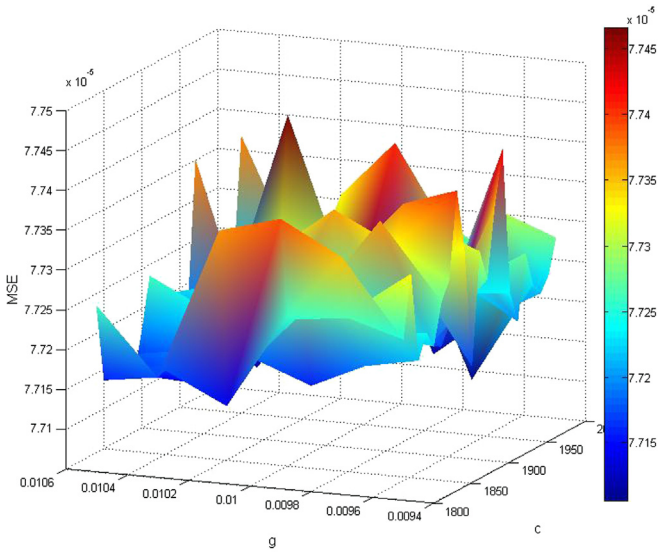

Fig. 9. Second step search for c and g among the smaller region of C and G, which are found in the rough search.

4. Conclusions

In this paper, we utilized the SVR to estimate SOC for electric vehicles under diversified driving conditions and found the optimal parameters for the SVR based on double step searches. The performance of the SOC estimation using SVR can be efficiently improved by selecting optimal cost factor c and parameter g in the RBF kernel of the SVR. Double step search reduces the time for the training process of SVR and avoids searching the parameters in a large range blindly. In this way, it enables us to save lots of recourses and speed up the experimental process. The proposed SVR addresses the SOC prediction problems with the following advantages: 1) the double search process for parameters in SVR speeds up the training process, while the training process for ANN is time-consuming; 2) the architecture of SVR is simple and elegant, while it is usually difficult to determine the architecture of ANN; 3) the SVR can be scalable to integrate knowledge from other indicators such as temperature, power, etc., while the traditional OCV and Current integral methods are sensitive to the accurate measurement of the current and voltage. When it is evaluated using the cycle conditions in ADVISOR, the estimation proposed in this paper behaves more efficiently and accurately than the estimations based on ANN. Since the cycle conditions that ADVISOR provides are universal, our method is applicable to all kinds of batteries including lead-acid, metal hydride–nickel and lithium ion batteries. The future work for the SOC using SVR will be focused on evaluation for more complicated driving conditions and the improvement of the robustness of the SVR in real situations.

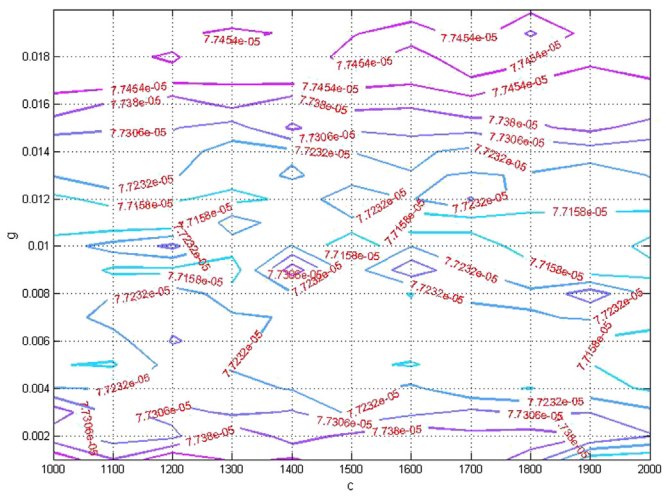

Fig. 8. MSE contour for rough search.

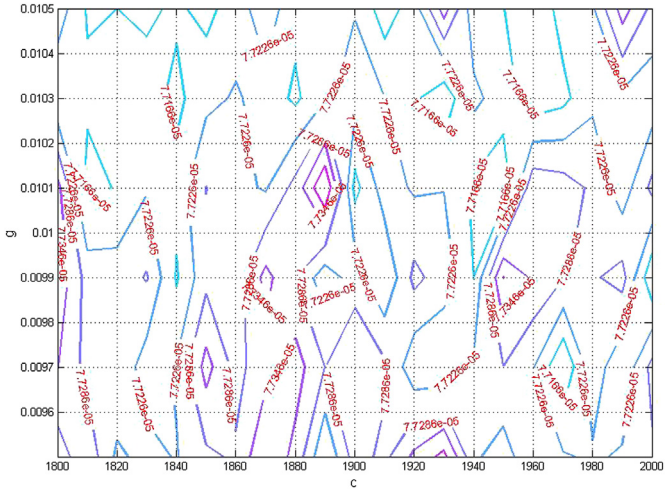

Fig. 10. MSE contour for second step search.

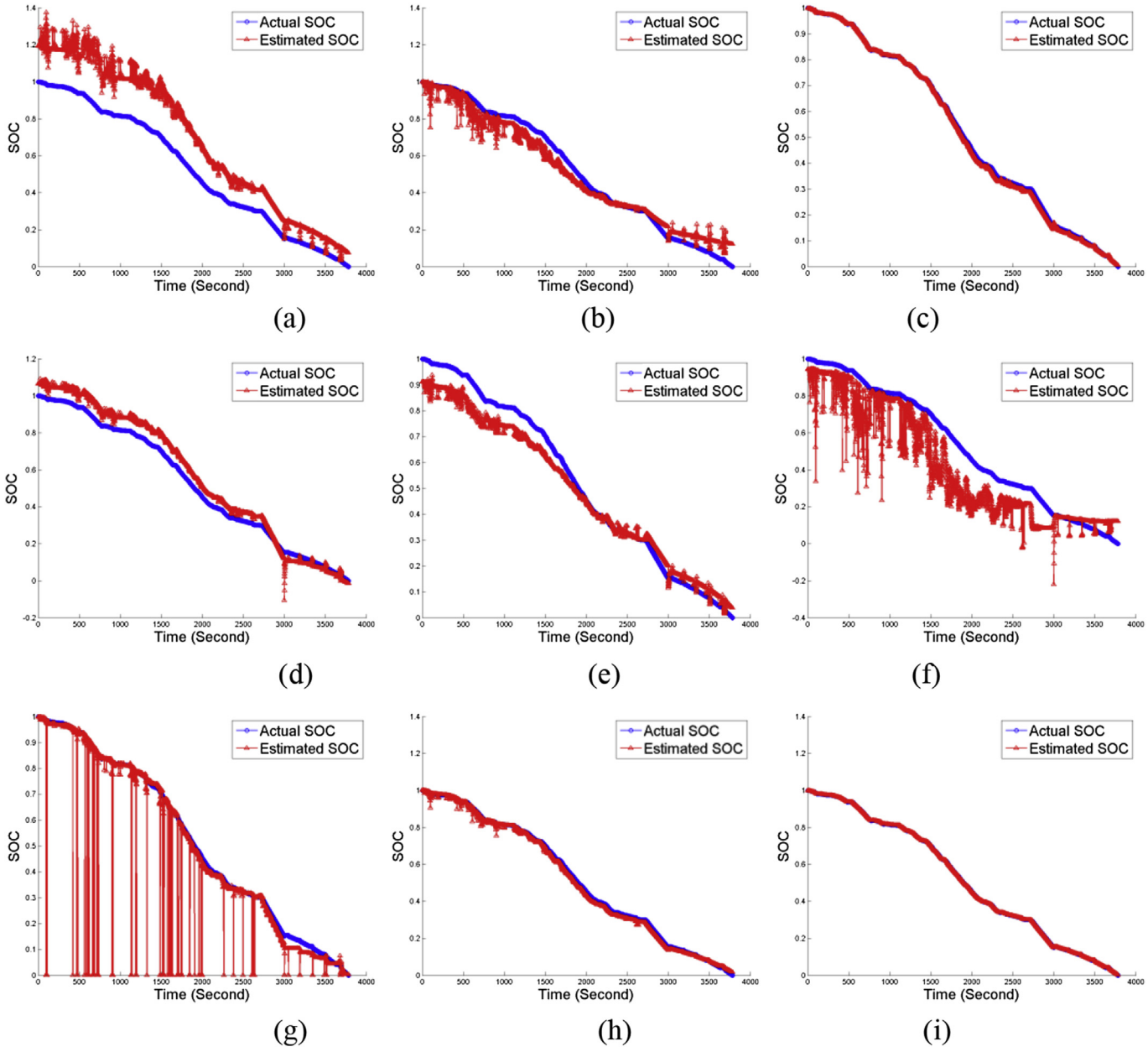

Fig. 11. Estimated SOC of combination 1 by (a) BP1; (b) BP2; (c) BP3; (d) DBP1; (e) DBP2; (f) ELMAN; (g) GRNN; (h) RBFNN; (i) SVR.

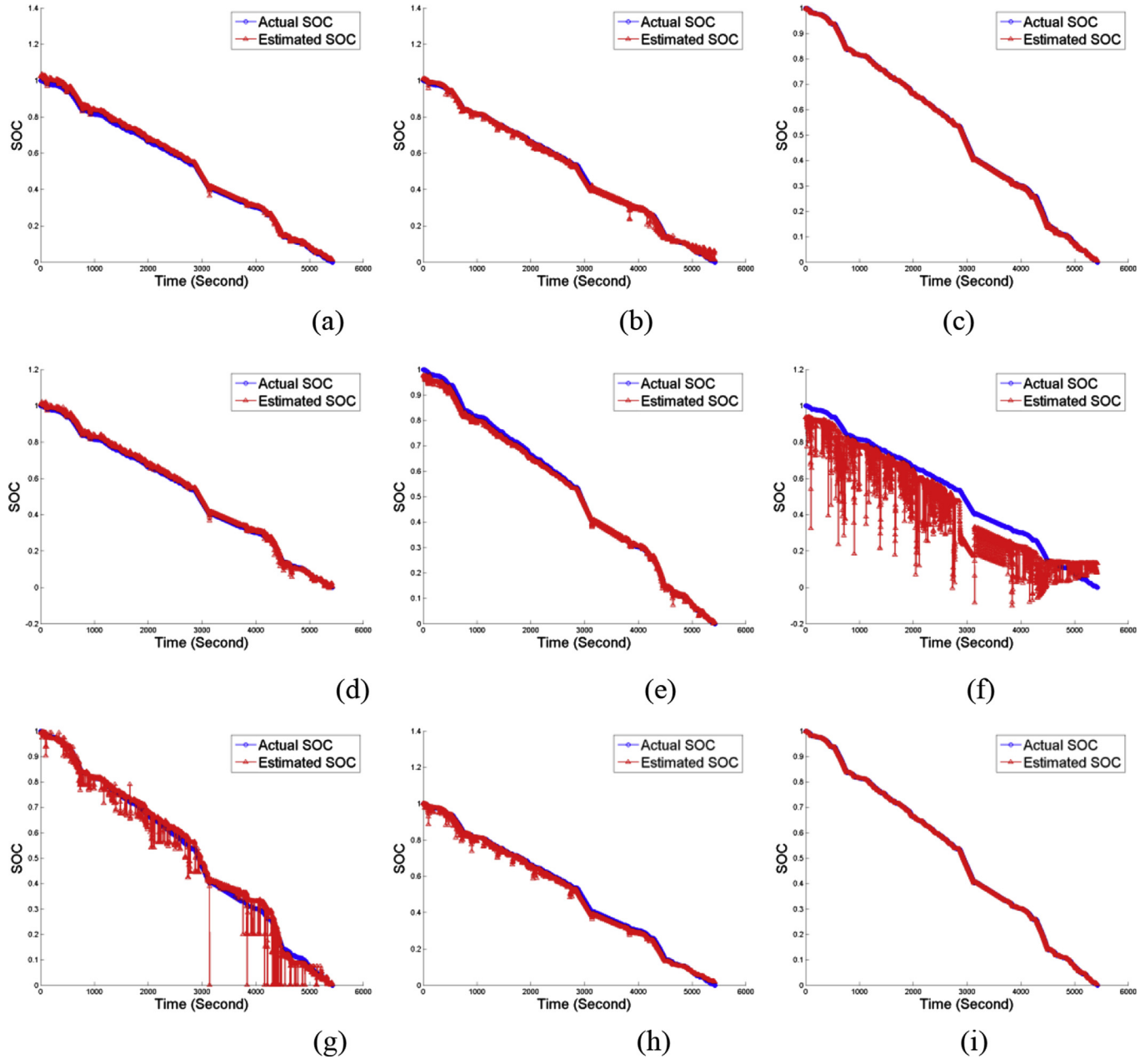

Fig. 12. Estimated SOC of combination 2 by (a) BP1; (b) BP2; (c) BP3; (d) DBP1; (e) DBP2; (f) ELMAN; (g) GRNN; (h) RBFNN; (i) SVR.

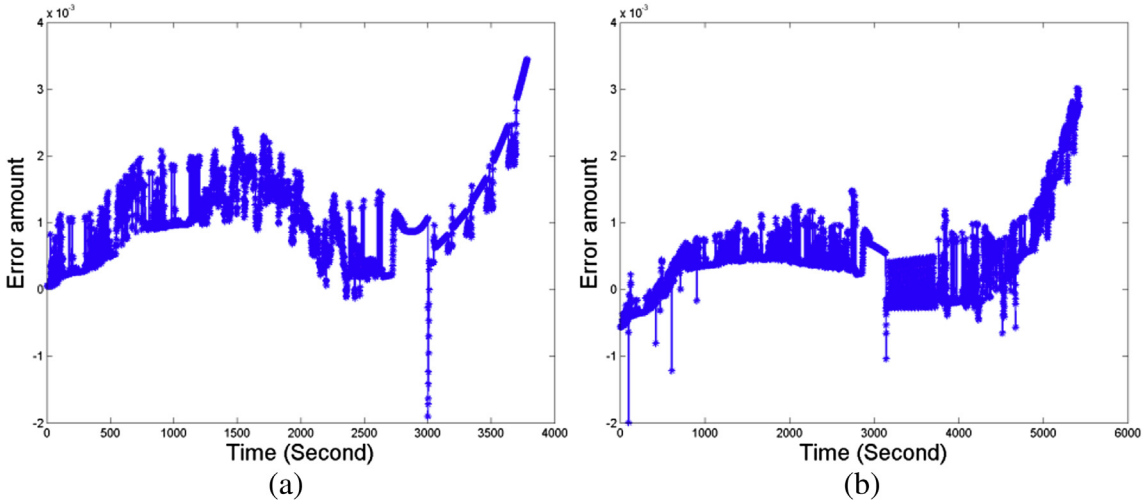

Fig. 13. (a) Combination 1; (b) Combination 2; error amount = actual SOC − estimated SOC.

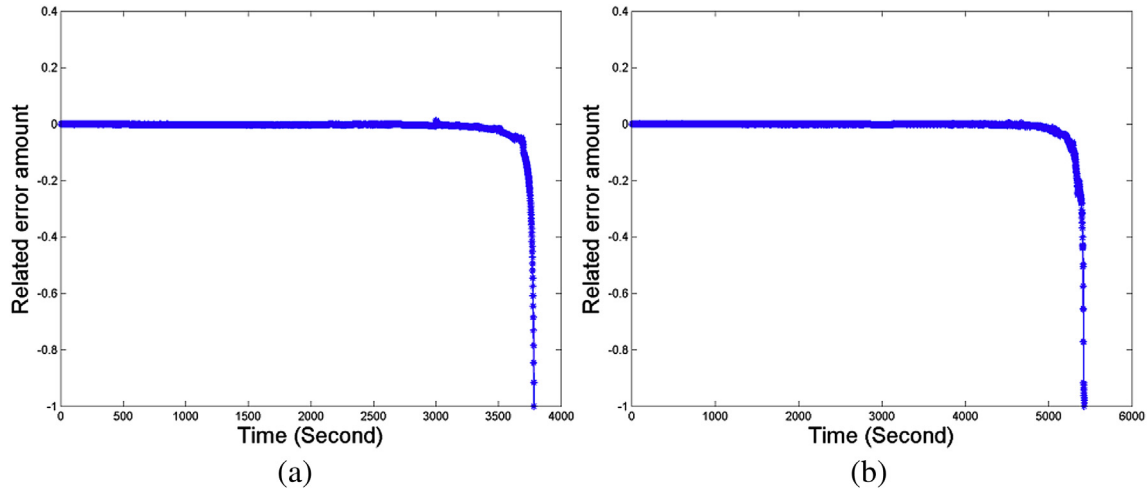

Fig. 14. (a) Combination 1; (b) Combination 2; relative error amount = (actual SOC − estimated SOC)/actual SOC.

Table 1
MSE for the 10 cycle conditions using 9 estimators.

Type	BP1 (5tansig)	BP2 (20tansig)	BP3 (5logsig)	DBP1 (5tansig+ 5logsig)	DBP2 (5tansig + 5tansig)	GRNN	ELMAN	RBFNN	SVM
Combination 1	350380.65	29684.85	453.16	36955.2	40472.85	166395.65	220549.65	1385.28	**15.49**
Combination 2	3452.6	784.58	68.06	1804.93	2092.37	10995.54	158719.57	1221.63	**5.56**
UDDSHDV	20473.78	1236.84	84.75	2919.74	4587.24	9431.41	98777.72	677.37	**3.93**
CSHVR	454.67	233.5	394.3	702.14	1113.52	4996.98	108792.81	841.82	**2.59**
COMMUTER	3085.85	43159.27	570.88	10033.17	3660.19	1651793.02	758497.83	17367.45	**40.07**
CBD14	235302.69	233307.47	222547.06	228574.22	226670.84	**173,439**	205935.41	207206.4	226917.58
WVUINTER	36261.95	4257.51	227.86	7091.14	10766.54	31,226	237956.76	914.32	**7.1**
5PEAK	17545.73	1301.93	387.77	2389.5	2206.68	29490.69	180639.4	2124.37	**12.17**
ARTERIAL	26043.62	24215.59	59.34	4959.77	4908.19	404360.24	554635.06	8196.61	**12.05**
NYCCOMP	422.71	100.47	177.41	561.15	526.24	1219.02	85676.35	731.94	**2.94**

Note: to display the numeric value clearly in the table, all the above results have been multiplied by 10^-7. The number using the bold font is the best result with respect to other estimators.

Table 2
Error sum for the 10 cycle conditions using 9 estimators.

Type	BP1 (5tansig)	BP2 (20tansig)	BP3 (5logsig)	DBP1 (5tansig + 5logsig)	DBP2 (5tansig + 5tansig)	GRNN	ELMAN	RBFNN	SVM
Combination 1	656.11	165.96	18.64	189.3	219.14	134.82	444.26	35.69	**4.05**
Combination 2	93.07	31.08	10.18	58.83	71.62	111.22	541.76	46.62	**2.94**
UDDSHDV	212.57	43.13	10.07	70.47	104.31	118.73	385.47	33.23	**2.44**
CSHVR	33.65	20.54	27.34	44.31	42.85	80.95	530.18	45.13	**2.42**
COMMUTER	30.67	76.69	13.17	61.65	35.97	724.93	607.96	88.89	**3.62**
CBD14	758.76	753.27	736.5	733.21	739.92	**587.15**	651.24	696.56	742.52
WVUINTER	150.09	42.63	9.88	61.2	86.39	113.36	347.74	21.5	**2.12**
5PEAK	219.44	46.35	25.57	70.92	66.58	194.53	673.51	70.18	**3.73**
ARTERIAL	190.32	131.29	6.78	77.58	78.74	467.65	857.77	100.48	**2.9**
NYCCOMP	50.03	18.93	26	56.13	40.05	70.34	712.1	62.63	**3.27**

Note: the number using the bold font is the best result with respect to other estimators.

Acknowledgement

The authors are highly grateful for the financial support from by the joint project of National Natural Science Foundation of China and Natural Science Foundation of Guangdong Province (Grant No. U1134002), the National Natural Science Foundation (Grant No. 21273084), the Natural Science Fund of Guangdong Province (Grant No. 10351063101000001), the key project of Science and Technology in Guangdong Province (Grant No. 2011A010801001), and the scientific research project of Department of Education of Guangdong Province (Grant No. 2013CXZDA013).

Appendix. A list of explanations on variances and abbreviations.

α_j	a positive value that measures the weight for the jth support vectors $\boldsymbol{x}_j$
b	a positive value that measures the threshold of the predicted function $f(\boldsymbol{x})$
c	the optimal cost factor in the RBF kernel of SVR
g	the variance of gamma parameter $g = -1/2\sigma^2$ in the RBF kernel of SVR
$\boldsymbol{x}$	the input data in m-dimensional space
$\boldsymbol{x}_j$	the jth support vector

y_j the corresponding output value of the jth support vectors $\boldsymbol{x}_j$
$\boldsymbol{w}$ the weight of the hyperplane in n-dimensional space
σ the standard deviation of the Gaussian kernel
ε the relaxed threshold of the error between the predicted value and the true value
ξ_i, ξ_i^* the slack variables for the optimization problem
C_n the normal charge
C_r the remaining charge
(C, G) the optimal pairs in the RBF kernel of SVR
$\mathbb{R}^m$ the input space
$\mathbb{R}^n$ the high dimensional space
$\mathscr{H}$ the Reproducing Kernel Hilbert Space (RKHS)
ADVISOR advanced vehicle simulator
ANN artificial neural network
ARTERIAL the cycle condition of arterial road
BMS battery management system
BPNN back propagation neural network
CBD14 the cycle condition in the central business district
COMMUTER the cycle condition by Commuter Cars
CSHVR the cycle condition by heavy vehicle in city suburb
ESS energy storage system
GRNN generalized regression neural network
MSE mean square error
NETSUM the weighted netsum function
NYCCOM the cycle condition of community road in New York City
OCV open circuit voltage measurement
5PEAK the cycle condition in the peak-hour traffic
RBF radial basis function kernel
RBFNN radial basis function neural network
RKHS Reproducing Kernel Hilbert Space
SOC state of charge
SVR support vector machine for regression
UDDSHDV the cycle condition by heavy duty vehicle in Urban Dynamometer Driving Schedule
WVUINTER the cycle condition of intercontinental road researched by West Virginia University

References

[1] N.A. Monfared, N. Gharib, H. Moqtaderi, M. Hejabi, M. Amiri, F. Torabi, A. Mosahebi, J. Power Sources 158 (2006) 932–935.
[2] B.Z. Li, Y. Wang, L. Xue, X.P. Li, W.S. Li, J. Power Sources 232 (2013) 12–16.
[3] J. Yi, X.P. Li, S.J. Hu, W.S. Li, L. Zhou, M.Q. . Xu, J.F. . Lei, L.S. . Hao, J. Power Sources 196 (2011) 6670–6675.
[4] Y. Saito, M. Shikano, H. Kobayashi, J. Power Sources 196 (2011) 6889–6892.
[5] J. Gomez, R. Nelson, E.E. Kalu, M.H. Weatherspoon, J.P. Zheng, J. Power Sources 196 (2011) 4826–4831.
[6] Z. Chen, C.C. Mi, Y.H. Fu, J. Xu, X.Z. Gong, J. Power Sources 240 (2013) 184–192.
[7] J.S. Wang, E. Sherman, M. Verbrugge, P. Liu, J. Power Sources 196 (2011) 9648–9653.
[8] D.Y. Zhou, G.Z. Wang, W.S. Li, G.L. Li, C.L. Tan, M.M. Rao, Y.H. Liao, J. Power Sources 184 (2008) 477–480.
[9] Y.H. Liao, D.Y. Zhou, M.M. Rao, W.S. Li, Z.P. Cai, Y. Liang, C.L. Tan, J. Power Sources 189 (2009) 139–144.
[10] N. Khare, P. Singh, J.K. Vassiliou, J. Power Sources 218 (2012) 462–473.
[11] G.L. Plett, J. Power Sources 134 (2004) 277–292.
[12] G.L. Plett, J. Power Sources 161 (2006) 1356–1368.
[13] G.L. Plett, J. Power Sources 134 (2006) 1369–1384.
[14] J. Lee, O. Nam, B.H. Cho, J. Power Sources 174 (2007) 9–15.
[15] Y.J. Xing, W.H. He, M. Pecht, K.L. Tsui, Appl. Energy 113 (2014) 106–115.
[16] K.S. Ng, C.S. Moo, Y.P. Chen, Y.C. Hsieh, Appl. Energy 86 (2009) 1506–1511.
[17] Y. He, X.T. Liu, C.B. Zhang, Z.H. Chen, Appl. Energy 101 (2013) 808–814.
[18] H.F. Dai, X.Z. Wei, Z.C. Sun, J.Y. Wang, W.J. Gu, Appl. Energy 95 (2012) 227–237.
[19] X.S. Hu, S.B. Li, H. Peng, F.C. Sun, J. Power Sources 217 (2012) 209–219.
[20] T. Weigert, Q. Tianb, K. Lianb, J. Power Sources 196 (2011) 4061–4066.
[21] S.J. Ciou, K.Z. Fung, K.W. Chiang, J. Power Sources 175 (2008) 338–344.
[22] C. .Cortes, V. .Vapnik, Mach. Learn. 20 (1995).
[23] T. Hansen, C.J. Wang, J. Power Sources 141 (2005) 351–358.
[24] D. Andre, C. Appel, T. Soczka-Guth, D.U. Sauer, J. Power Sources 224 (2013) 20–27.
[25] A.J. Smola, B. Schölkopf, Stat. Comp. 14.3 (2004) 199–222.
[26] V. .Cherkassky, Y.Q. . Ma, Neural Networks 17 (2004) 112–126.
[27] M. Kindelan, V. Bayona, Eng. Anal. Bound. Elem. 37 (2013) 1617–1624.